




# Ten Years of Hunting for Similar Code for Fun and Profit

**Stéphane Glondu**

**Lingxiao Jiang**

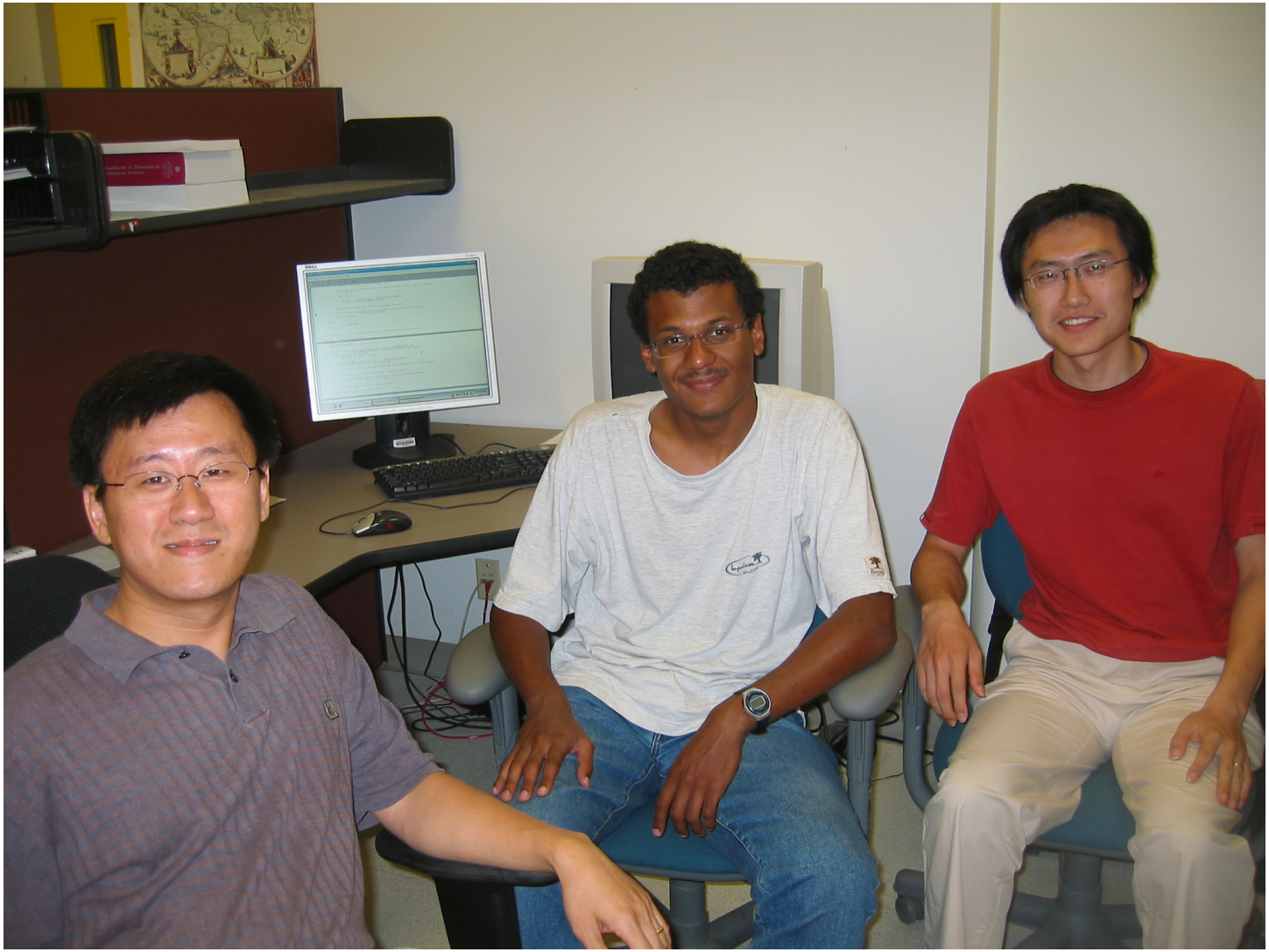
**Zhendong Su**

Software is about similarity!

# DECKARD: Scalable and Accurate Tree-based Detection of Code Clones*

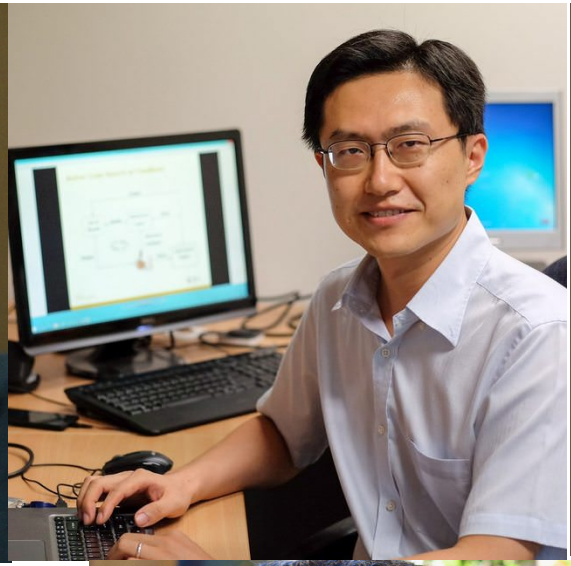
Lingxiao Jiang Ghassan Misherghi Zhendong Su
*University of California, Davis*
{lxjiang,ghassanm,su}@ucdavis.edu

Stéphane Glondu
*ENS de Cachan, France*
steph@glondu.net



2005

2006

# But, what is impact?

# Concept

# Insight

# Technique

# Tool

# Software eats everything

Software is about similarity

# Algorithms

# Procedural Abstraction

Algorithms

Procedural Abstraction

Modularity

Algorithms

Procedural Abstraction

Modularity

Design Patterns

Algorithms

Procedural Abstraction

Modularity

Design Patterns

Object Orientation

Algorithms

Procedural Abstraction

Modularity

Design Patterns

Object Orientation

Algorithms

Refactoring

Procedural Abstraction

Modularity

Design Patterns

Object Orientation

Algorithms

Refactoring

Software Analytics

Procedural Abstraction

Modularity

Design Patterns

Object Orientation

Algorithms

Refactoring

Software Analytics

“Big Code”

# How to define similarity?

# Textual

# Token

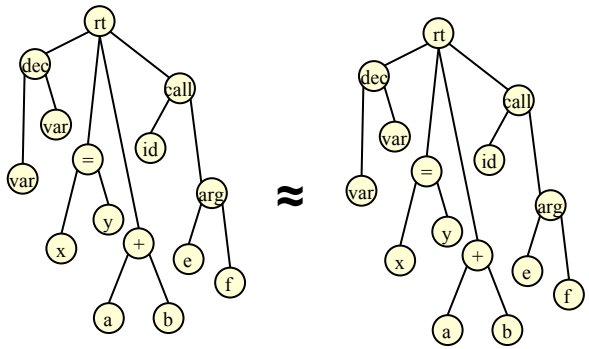
# AST

PDG

# Functional

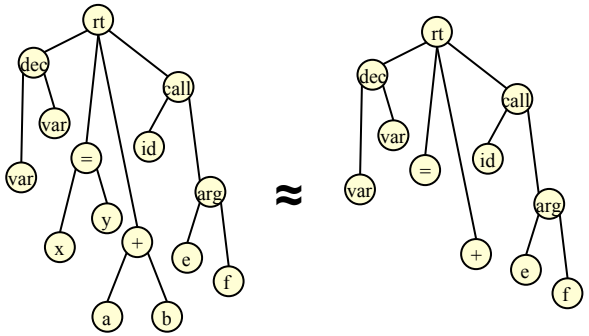
# Deckard

# Goal



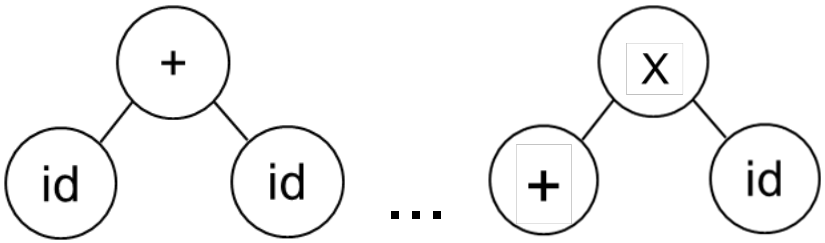

*Scalable tree similarity*

# Concept



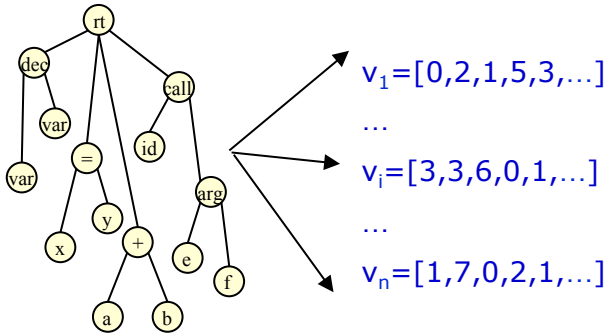

*Approximate*
*tree similarity*

# Insight



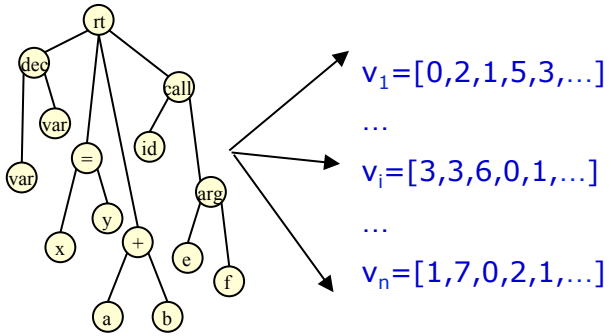

*q-level subtrees*

# Technique



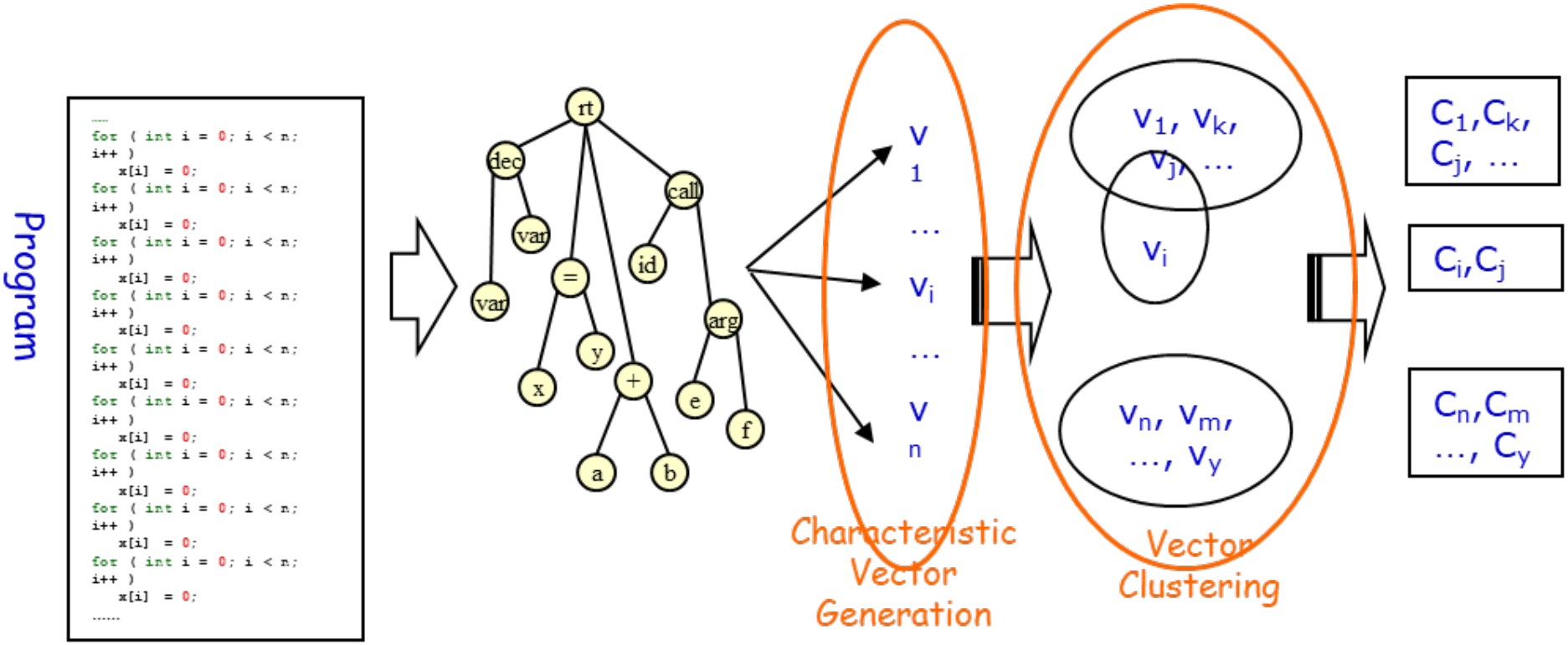

*Vector embedding of trees*

# Technique




*Vector embedding of trees*

*Locality sensitive hashing*


Program
.....
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
for ( int i = 0; i < n;
i++ )
x[i] = 0;
.....
rt
dec
var
var
=
call
id
arg
x
y
+
e
f
a
b
v 1
...
v i
...
v n
Characteristic Vector Generation
v1, vk, vj, ...
vi
vn, vm, ..., vy
Vector Clustering
C1,Ck, Cj, ...
Ci,Cj
Cn,Cm ..., Cy

# Tool


*Deckard*
(on GitHub)

# Follow-up

*ISSTA'09*, July 19–23, 2009, Chicago, Illinois, USA

# Detecting Code Clones in Binary Executables*

Andreas Sæbjørnsen
University of California, Davis
andsebjo@ucdavis.edu

Jeremiah Willcock†
Indiana University
jewillco@osl.iu.edu

Thomas Panas
Lawrence Livermore National Laboratory
panas2@llnl.gov

Daniel Quinlan
Lawrence Livermore National Laboratory
dquinlan@llnl.gov

Zhendong Su
University of California, Davis
su@ucdavis.edu

ESEC-FSE'07: Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering: Dubrovnik, Croatia, September 3-7, 2007, Pages 55-64.
http://doi.org/10.1145/1287624.1287634

# Context-Based Detection of Clone-Related Bugs*

Lingxiao Jiang Zhendong Su Edwin Chiu
Department of Computer Science
University of California, Davis
{lxjiang, su, eychiu}@ucdavis.edu

*OOPSLA/SPLASH'10*, October 17–21, 2010, Reno/Tahoe, Nevada, USA

# Scalable and Systematic Detection of Buggy Inconsistencies in Source Code

Mark Gabel[1] Junfeng Yang[2*] Yuan Yu[3] Moises Goldszmidt[3] Zhendong Su[1†]

[1] University of California, Davis
{mggabel, su}@ucdavis.edu

[2] Columbia University
junfeng@cs.columbia.edu

[3] Microsoft Research, Silicon Valley
{yuanbyu, moises}@microsoft.com

*FSE* '14, November 16–22, 2014, Hong Kong, China

# Vector Abstraction and Concretization for Scalable Detection of Refactorings

Narcisa Andreea Milea
School of Computing
National University of Singapore
mileanar@comp.nus.edu.sg

Lingxiao Jiang
School of Information Systems
Singapore Management University
lxjiang@smu.edu.sg

Siau-Cheng Khoo
School of Computing
National University of Singapore
khoosc@nus.edu.sg

*ICSE'08*, May 10–18, 2008, Leipzig, Germany

# Scalable Detection of Semantic Clones*

Mark Gabel Lingxiao Jiang Zhendong Su

Department of Computer Science
University of California, Davis
{mggabel,lxjiang,su}@ucdavis.edu

ISSTA '09: Proceedings of the 2009 International Symposium on Software Testing and Analysis, Chicago, July 19-23, 2009, Pages 81-92.
http://doi.org/10.1145/1572272.1572283

# Automatic Mining of Functionally Equivalent Code Fragments via Random Testing*

Lingxiao Jiang
University of California, Davis
jiangl@cs.ucdavis.edu

Zhendong Su
University of California, Davis
su@cs.ucdavis.edu

*FSE-18*, November 7–11, 2010, Santa Fe, New Mexico, USA

# A Study of the Uniqueness of Source Code*

Mark Gabel Zhendong Su

Department of Computer Science
University of California at Davis
{mggabel,su}@ucdavis.edu

ICSE 2012, Zurich, Switzerland

# On the Naturalness of Software

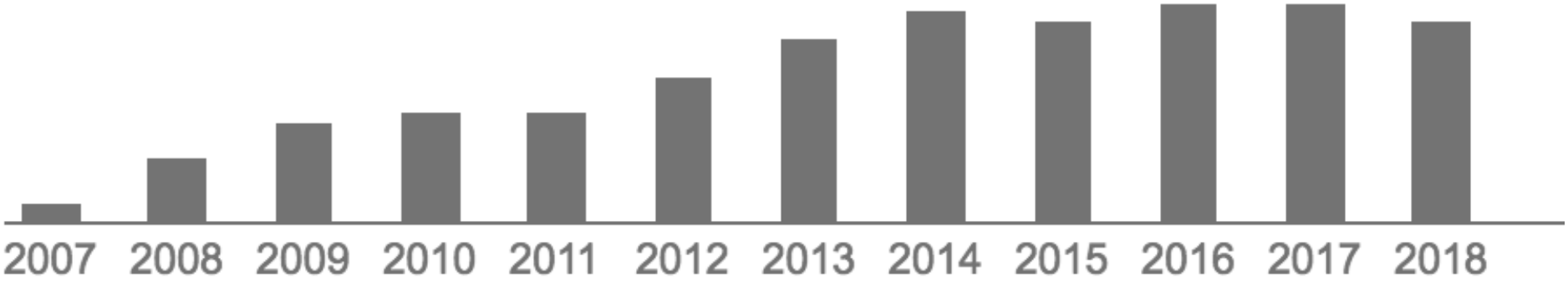
Abram Hindle, Earl T. Barr, Zhendong Su
*Dept. of Computer Science*
*University of California at Davis*
*Davis, CA 95616 USA*
*{ajhindle,barr,su}@cs.ucdavis.edu*

Mark Gabel
*Dept. of Computer Science*
*The University of Texas at Dallas*
*Richardson, TX 75080 USA*
*mark.gabel@utdallas.edu*

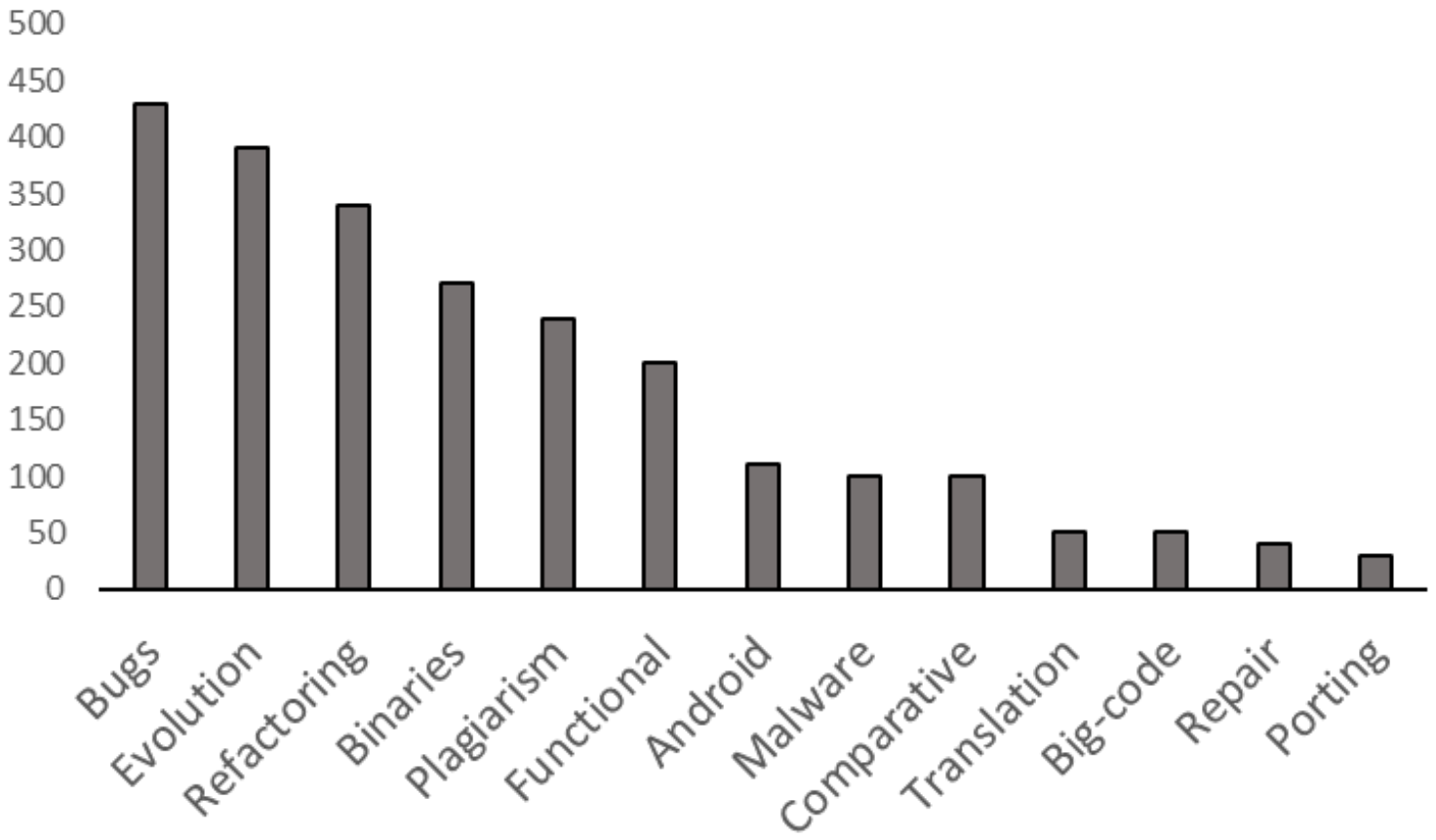
Premkumar Devanbu
*Dept. of Computer Science*
*University of California at Davis*
*Davis, CA 95616 USA*
*devanbu@cs.ucdavis.edu*

# Influence

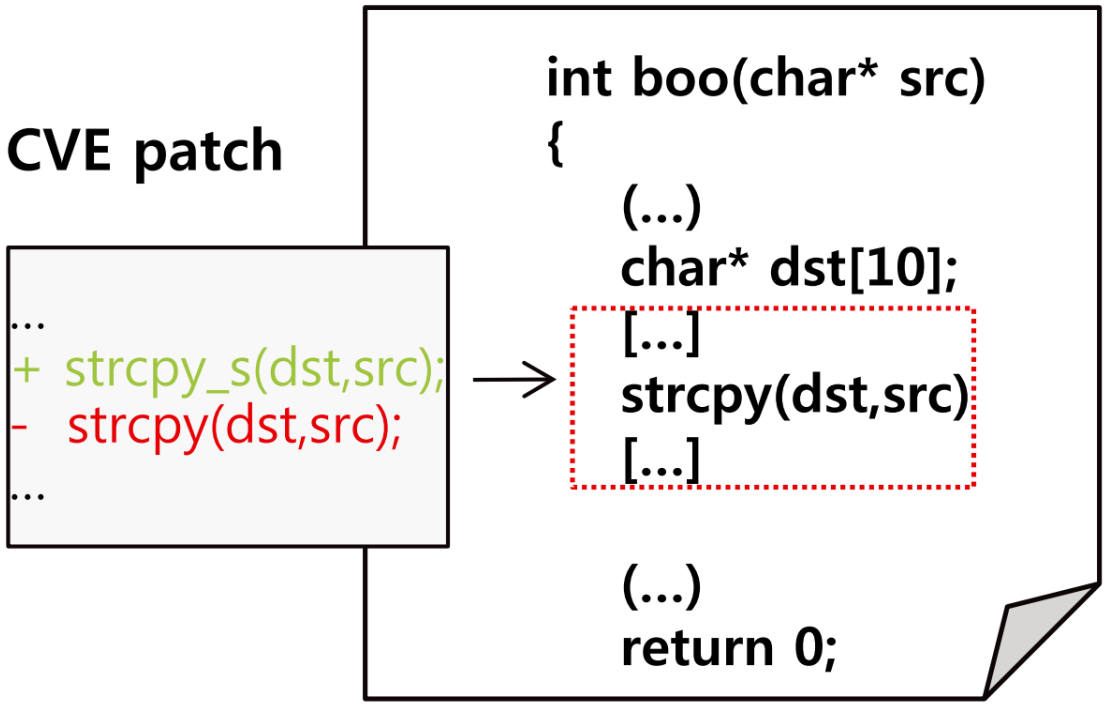
Cited by 671



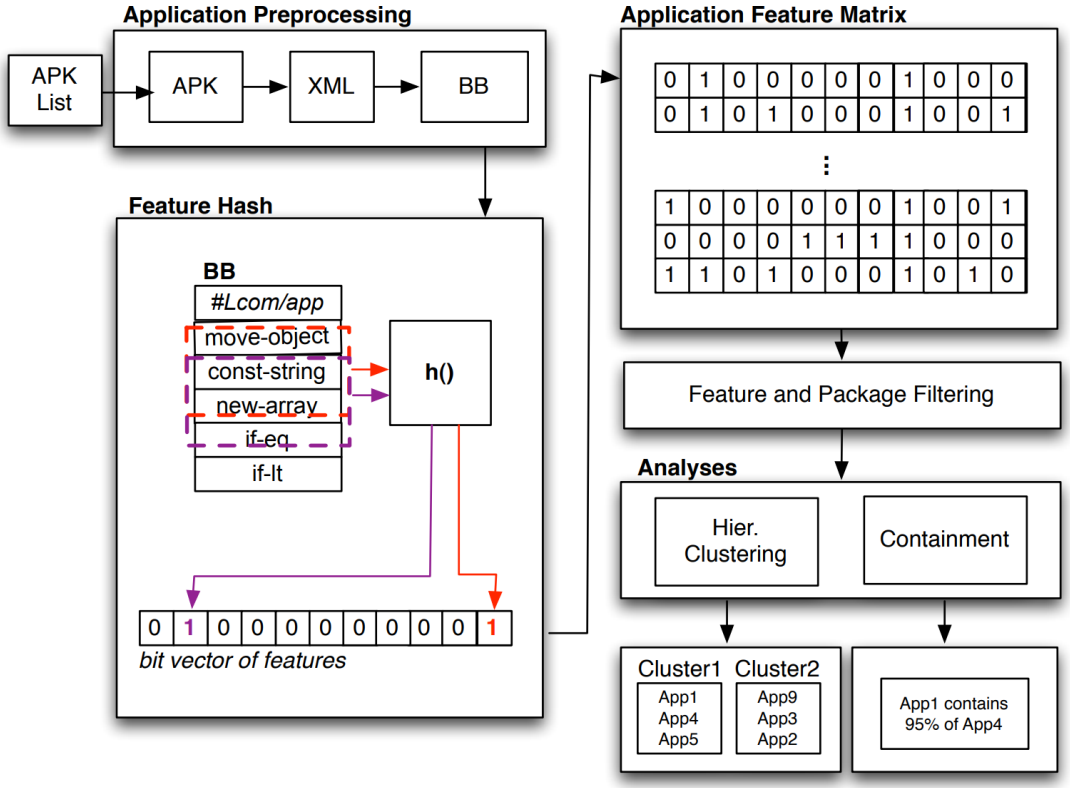
- 2008: Ours much more scalable, graph-based
- 2008: Lim et al., Static birthmarking
- 2008: Zhou et al., Combined approach
- 2008, Roy et al., NiCad
- 2009: Ours for similar code in binaries
- 2009: Juergens et al., ConQat
- 2009: Nguyen et al., ClemanX
- 2010: Lee et al., instant / intelligent code search
- 2010: Hummel et al., index-based
- 2010: Duala-Ekoko et al., clone tracking
- 2011: Tairas et al., sub-clones / clones in DSLs
- 2011: Göde et al., clone changes
- 2011: Higo et al., PDG clones
- 2011: Kim et al., MeCC
- 2012: Dang et al., XIAO clones
- 2012: Rahman et al., code clone smells
- 2012: Hanna et al, Juxtapp for Android apps
- 2012: Nguyen et al., JSync
- 2013: Khoo et al., Rendezvous for binaries
- 2013: Ng et al., Expose: binary clones
- 2013: Crussell et al., Andarwin for Android clones
- 2014: Hu et al., Duet for Android libraries
- 2014: Chen et al., Android clones
- 2014: Lin et al., clonepedia
- 2015: Wang et al., WuKong for Android apps
- 2015: Tian et al., birthmarks on dynamic seqs
- 2015: Wong et al., clocom: clones for comments
- 2016: Sajnani et al., SourcererCC
- 2016: White et al., deep learning for clones
- 2016: Su et al., code relatives
- 2017: Li et al., LibD for Android
- 2017: Cheng et al., cross-languages clones in diffs
- 2017: Palomba et al., code smells
- 2018: Kim et al., FaCoY
- 2018: Ragkhitwetsagul et al., image-based
- 2018: Wang et al., CCAligner
- 2018: Saini et al., Oreo
- 2018: Zhao et al., DeepSim


500
450
400
350
300
250
200
150
100
50
0
Bugs
Evolution
Refactoring
Binaries
Plagiarism
Functional
Android
Malware
Comparative
Translation
Big-code
Repair
Porting

# Bug Detection



# Malware Detection



# Refactoring

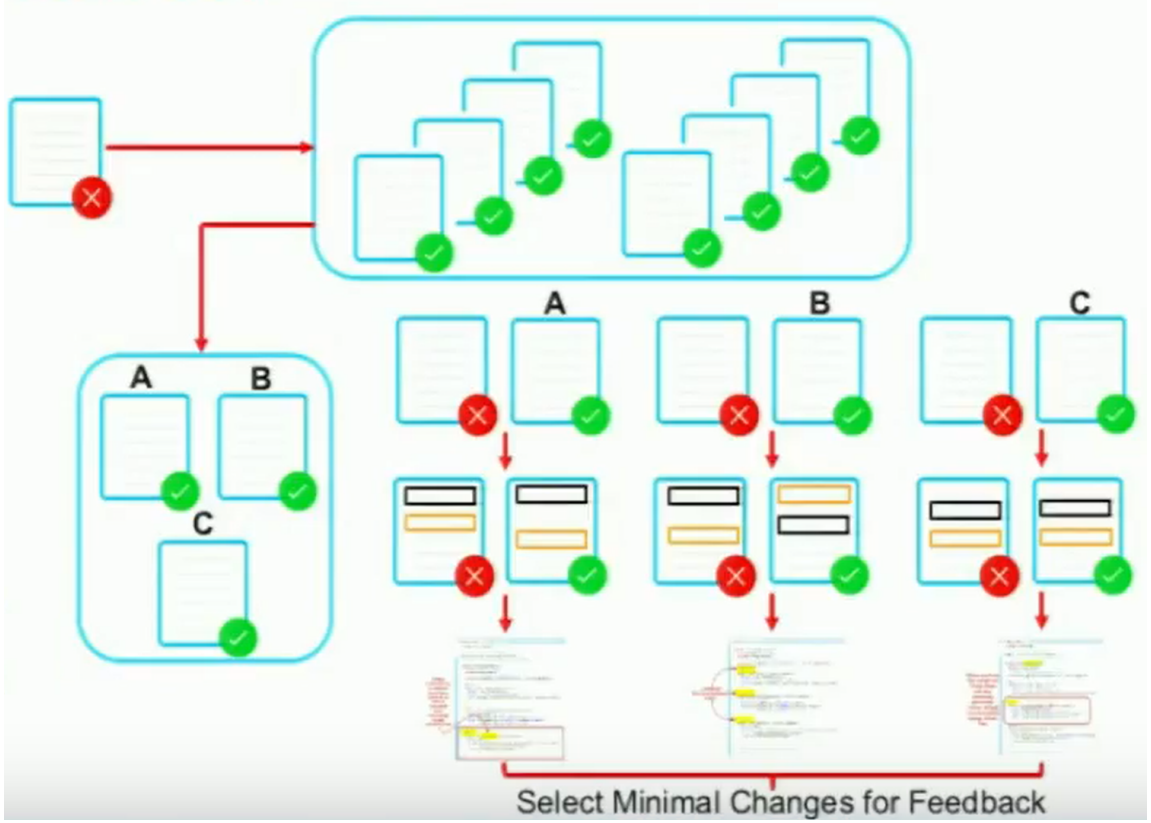
```
-    if (fContainer == null) {
+    IActionBars actionBars = fContainer.getActionBars();
+    if (actionBars==null&&offset!=0&&!fContainerProvided){
       return Utilities.findActionBars(fComposite, offset);
     }
-    return fContainer.getActionBars();
+    return actionBars;
   }
   public ISLocator getServiceLocator (int offset2) {
-    if (fContainer == null) {
+    ISLocator sLocator = fContainer.getServiceLocator();
+    if(sLocator == null&&offset2!=0&&!fContainerProvided){
       return Utilities.findSite(fComposite, offset2);
     }
-    return fContainer.getServiceLocator();
+    return sLocator;
```

# Refactoring

```
-    if (fContainer == null) {
+    IActionBars actionBars = fContainer.getActionBars();
+    if (actionBars==null&&offset!=0&&!fContainerProvided){
       return Utilities.findActionBars(fComposite, offset);
     }
-    return fContainer.getActionBars();
+    return actionBars;
   }
  public ISLocator getServiceLocator (int offset2) {
-    if (fContainer == null) {
+    ISLocator sLocator = fContainer.getServiceLocator();
+    if(sLocator == null&&offset2!=0&&!fContainerProvided){
       return Utilities.findSite(fComposite, offset2);
     }
-    return fContainer.getServiceLocator();
+    return sLocator;
```

```
T$0 v$0 = fContainer.m$0();
if (v$0==null && v$1!=0 && !fContainerProvided) {
  return Utilities.m$1(fComposite, v$1);
}
return v$0;
```

# Feedback Generation



# Reflections

# High-dimension vector similarity

Eureka moment
during a chat
with a friend

# High-dimension vector similarity

LSH was
not enough

# High-dimension vector similarity

Hack
internals

LSH was
not enough

# High-dimension vector similarity

Hack
internals

LSH was
not enough

Size-based
grouping

# High-dimension vector similarity

Hack
internals

LSH was
not enough

Size-based
grouping

Combine with
"normal" hashing

# Tree Representation

"Standard"
GCC front-end

# Tree Representation

"Standard"
GCC front-end

GCC
Gimple

# Tree Representation

"Standard"
GCC front-end

GCC
Gimple

Custom-built
parser generator

# False Positives

Probabilistic
Hashing

# False Positives

Probabilistic Hashing

q-level patterns

# False Positives

Probabilistic Hashing

q-level patterns

Relevance

# False Negatives

Probabilistic
Hashing

# False Negatives

Scope

Probabilistic Hashing

# False Negatives

Probabilistic Hashing

Scope

Granularity

# False Negatives

Probabilistic Hashing

Scope

Granularity

Semantic awareness

# Looking forward

# Is Deckard still relevant?

# Code Representation

Diff changes

Symbolic automata

Gated graphs

Complex network

# Notions of Similarity & Difference

Operational semantics

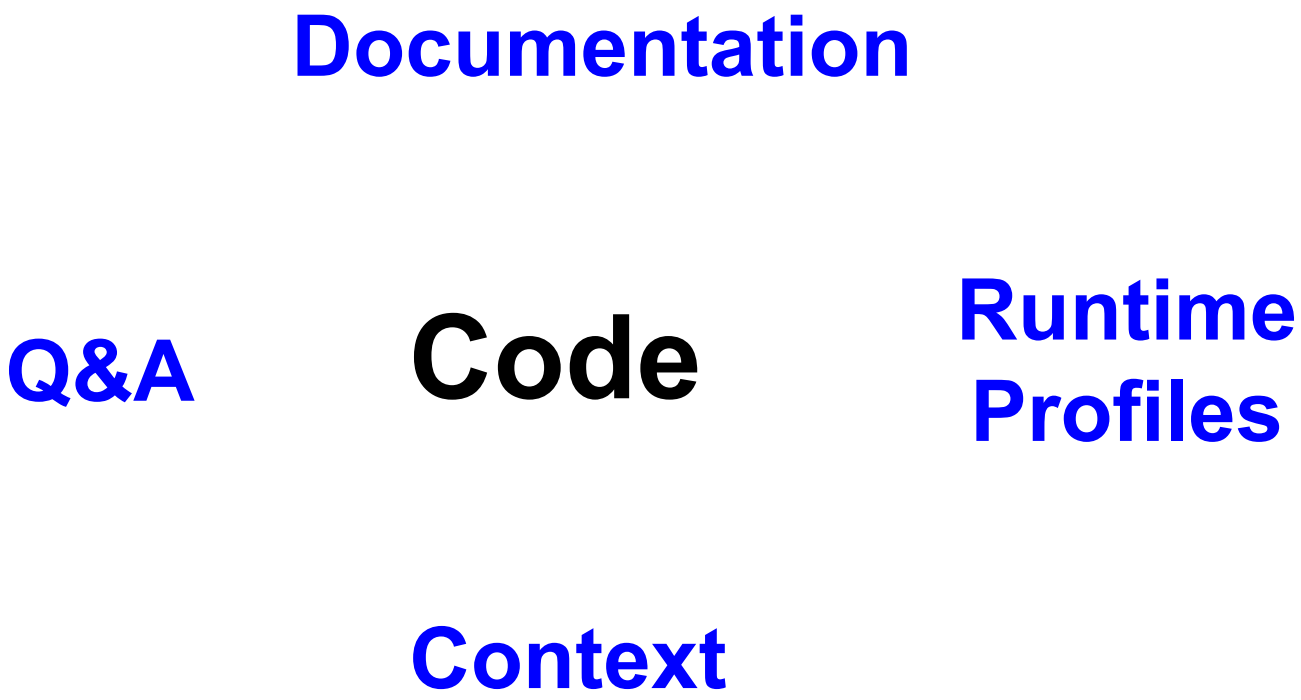
Input-Output

Specification

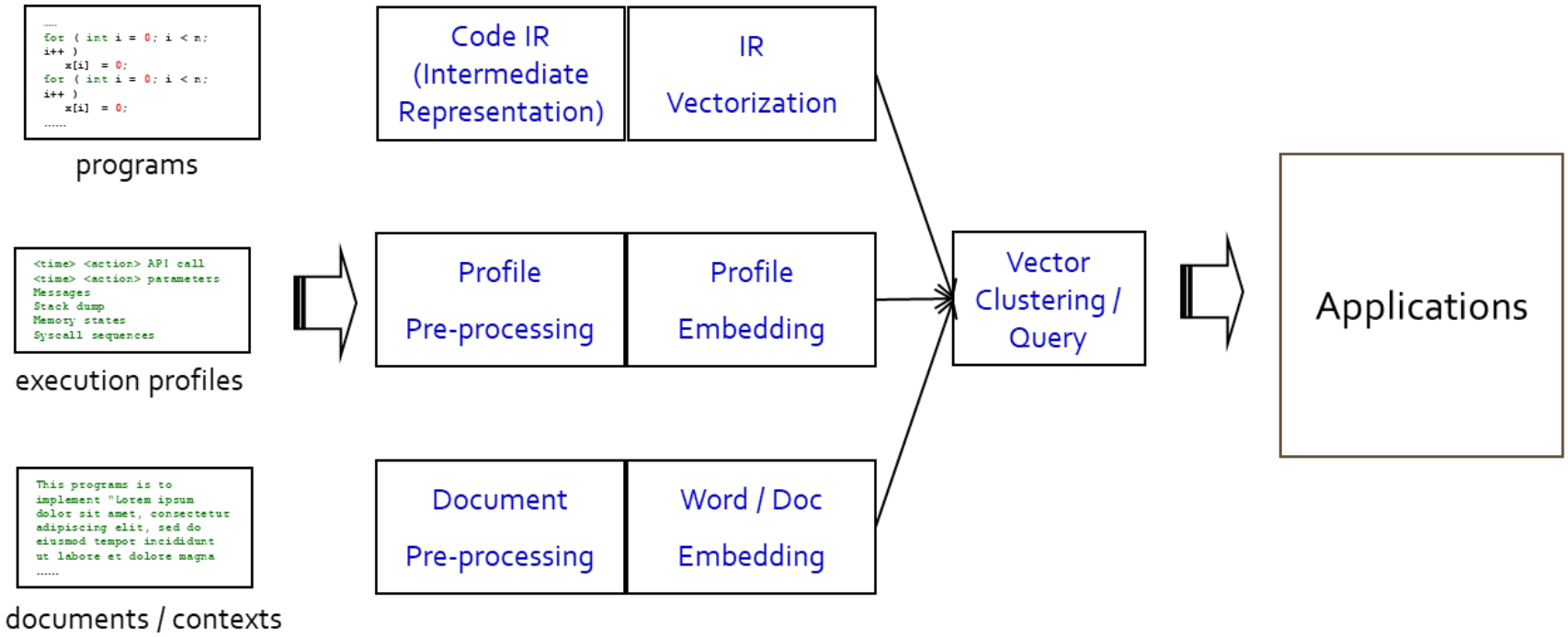
Description

# Beyond Code

# Code


Documentation
Q&A
Code
Runtime
Profiles
Context

# Generalization

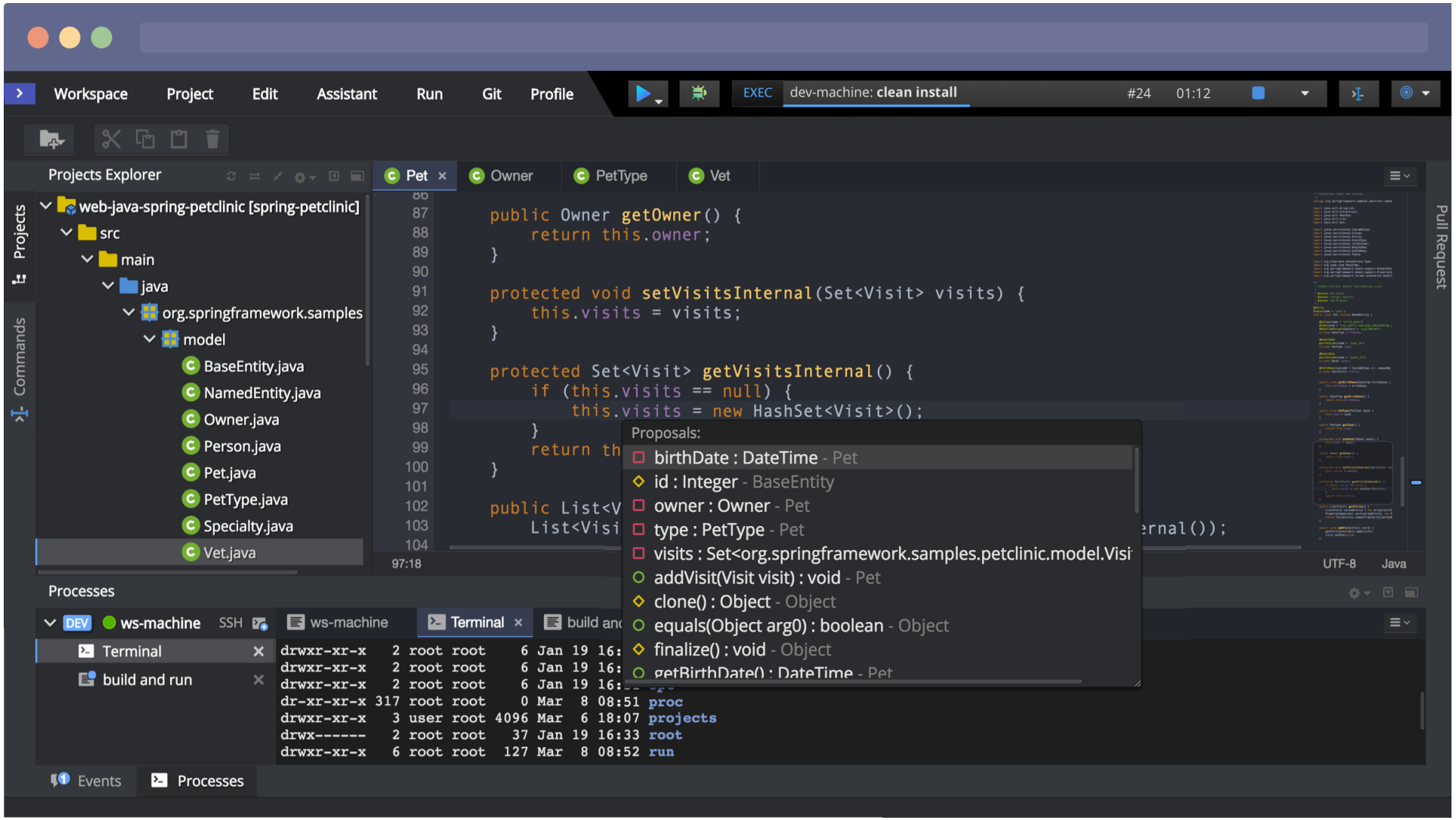


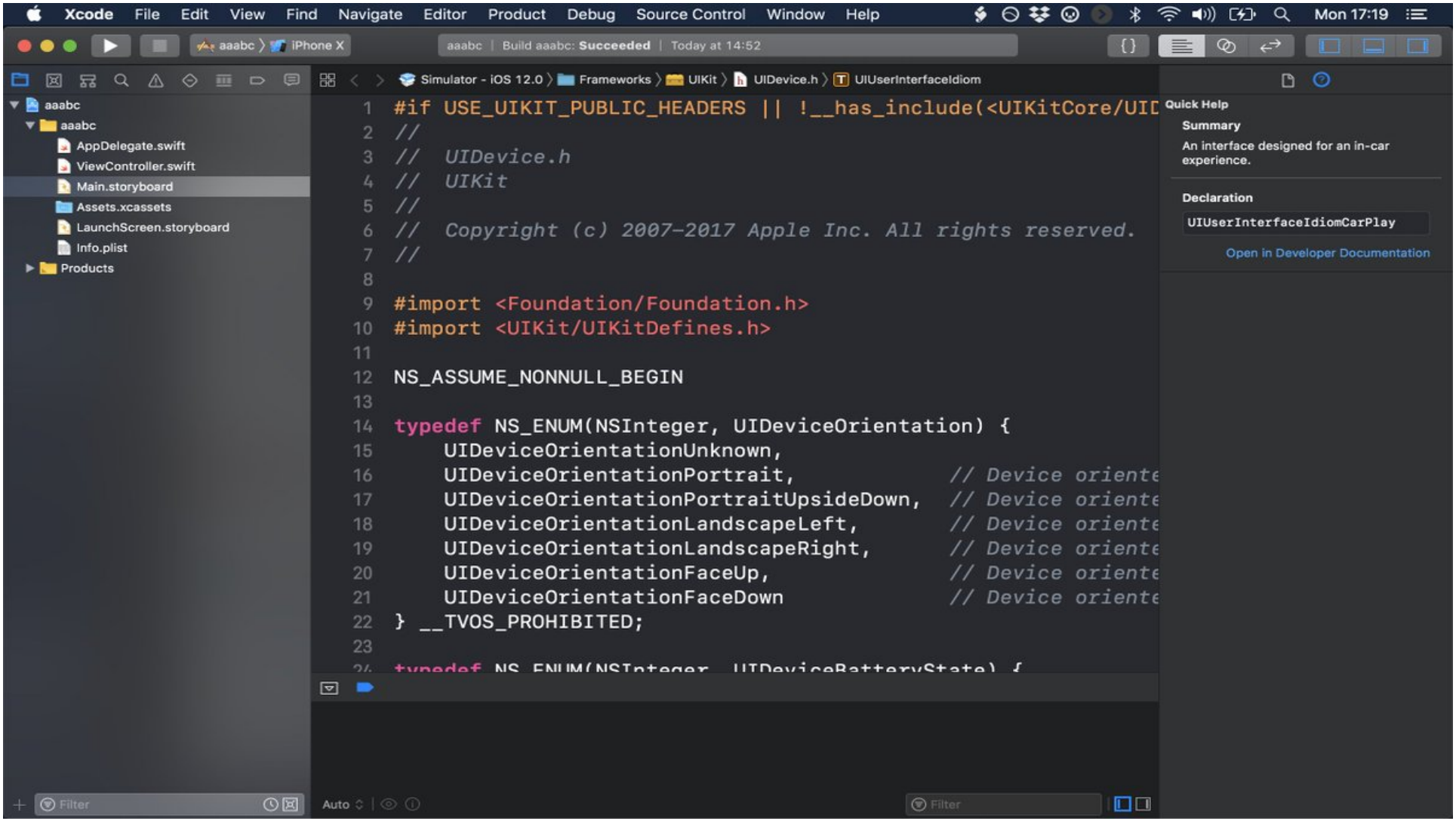
Looking much further

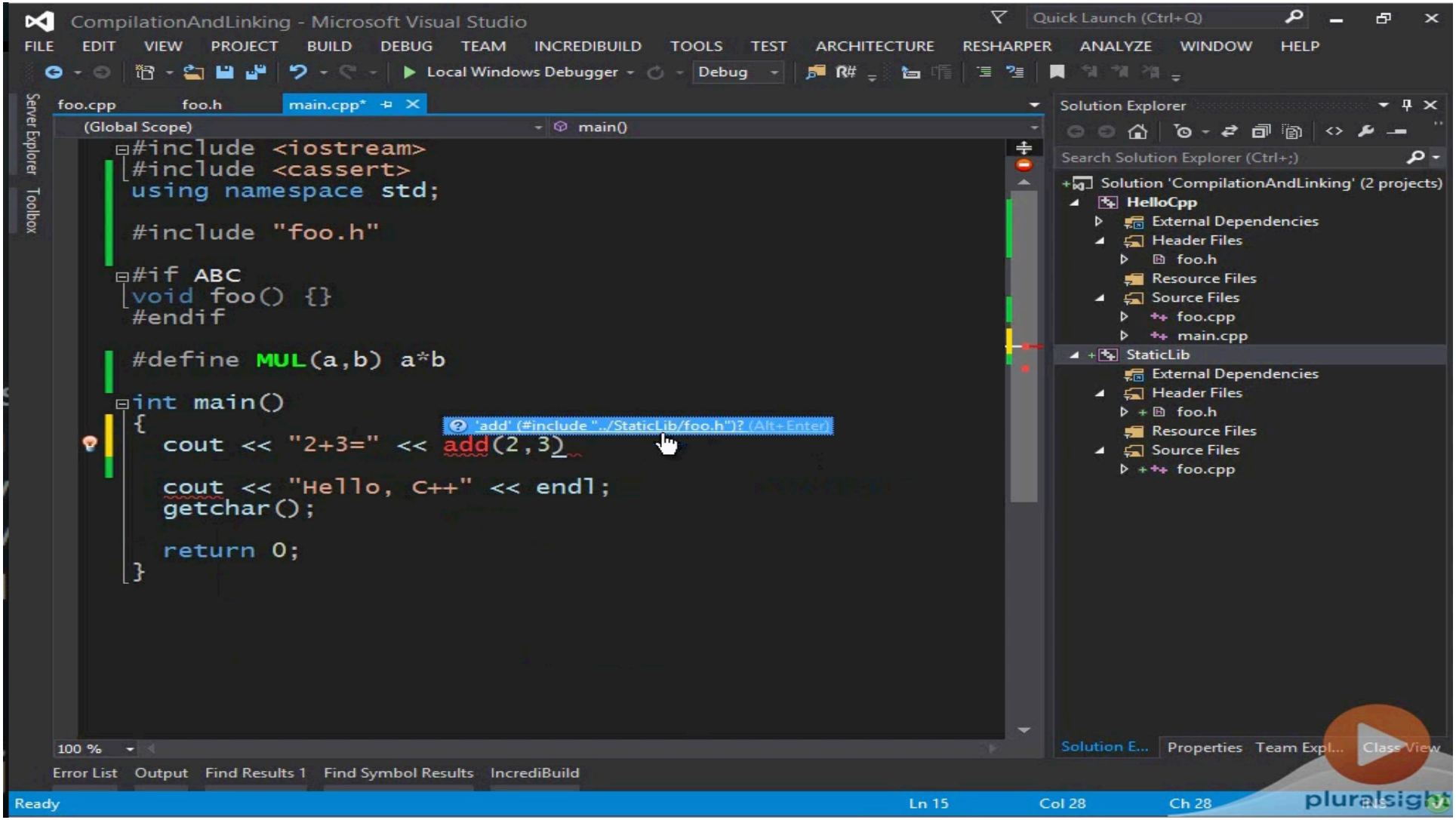

NATO Software Engineering
Conference 1968

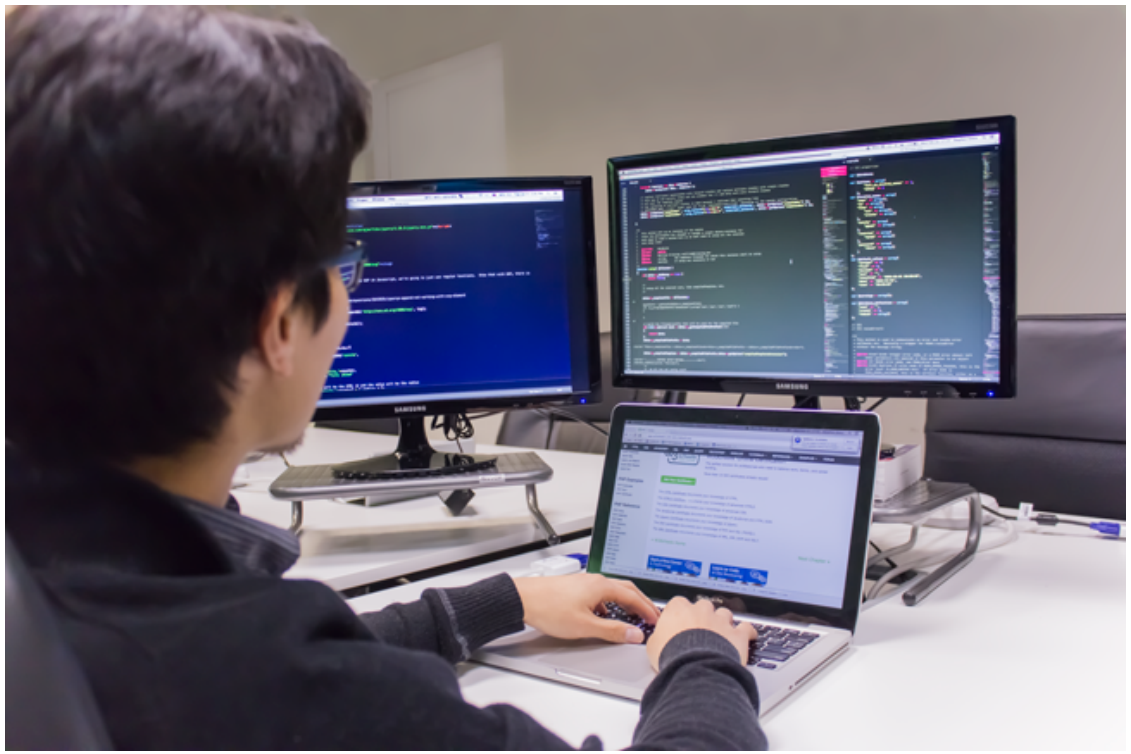
```c
#include <stdio.h>
#include <string.h>

int main(){
    char *word = "everest";
    char reverseword[strlen(word)+1];
    unsigned int letters_remaining = strlen(word);
    char *wordpointer = &word[strlen(word)-1];
    int i = 0;
    while(letters_remaining > 0){
        reverseword[i++] = *wordpointer--;
        letters_remaining--;
    }
    reverseword[strlen(word)] = '\0';
    printf("So the reversed word is %s\n",reverseword);
    return 0;
}
```

```
 1  #ifndef __ASM_ALPHA_FPU_H
 2  #define __ASM_ALPHA_FPU_H
 3
 4 !#include <asm/special_insns.h>
 5  #include <uapi/asm/fpu.h>
 6
 7  /* The following two functions don't need trapb/excb instructions
 8     around the mf_fpcr/mt_fpcr instructions because (a) the kernel
 9     never generates arithmetic faults and (b) call_pal instructions
10     are implied trap barriers.  */
11
12  static inline unsigned long
13  rdfpcr(void)
14  {
15      unsigned long tmp, ret;
16
17  #if defined(CONFIG_ALPHA_EV6) || defined(CONFIG_ALPHA_EV67)
18      __asm__ __volatile__ (
19          "ftoit $f0,%0\n\t"
20          "mf_fpcr $f0\n\t"
21          "ftoit $f0,%1\n\t"
22          "itoft %0,$f0"
23          : "=r"(tmp), "=r"(ret));
24  #else
25      __asm__ __volatile__ (
26          "stt $f0,%0\n\t"
27          "mf_fpcr $f0\n\t"
28          "stt $f0,%1\n\t"
29          "ldt $f0,%0"
30          : "=m"(tmp), "=m"(ret));
31  #endif
32
33      return ret;
34  }
35
36  static inline void
37  wrfpcr(unsigned long val)
38  {
39      unsigned long tmp;
40
41  #if defined(CONFIG_ALPHA_EV6) || defined(CONFIG_ALPHA_EV67)
42      __asm__ __volatile__ (
43          "ftoit $f0,%0\n\t"
44          "itoft %1,$f0\n\t"
.../linux/arch/alpha/include/asm/fpu.h  [Ins]                    ( 1,  0) [Top/1.8k] [
Eval: START
```

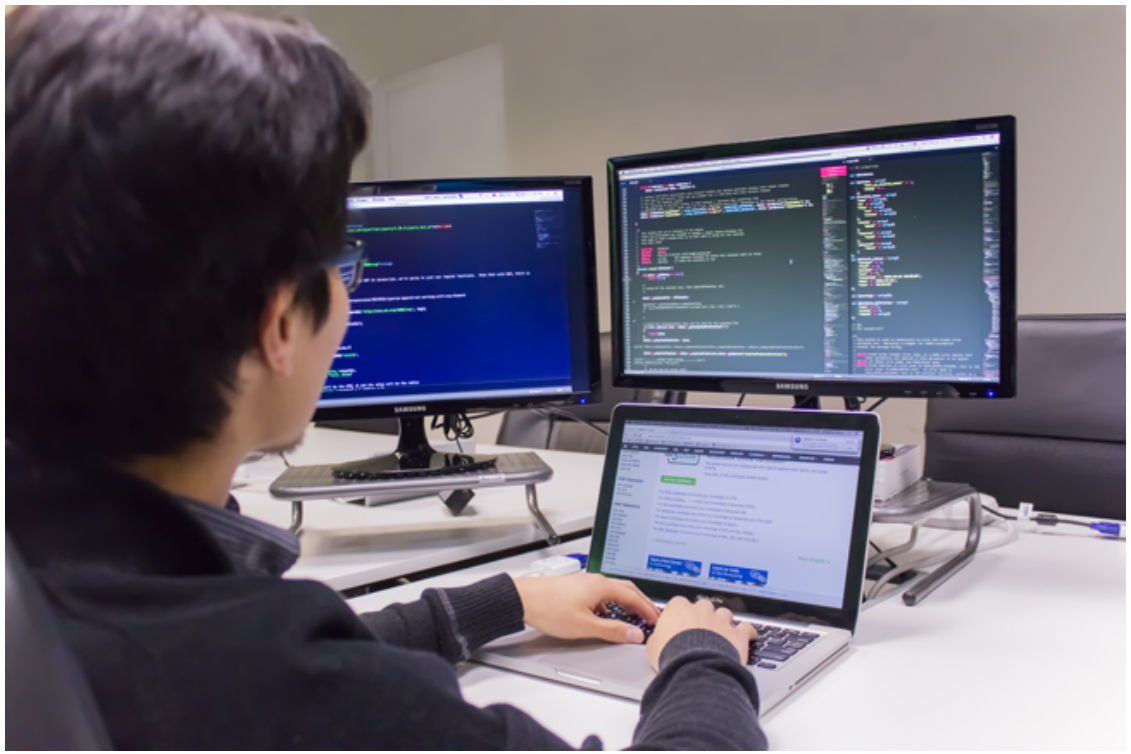

Workspace
Project
Edit
Assistant
Run
Git
Profile
EXEC
dev-machine: clean install
#24
01:12
Projects Explorer
web-java-spring-petclinic [spring-petclinic]
src
main
java
org.springframework.samples
model
BaseEntity.java
NamedEntity.java
Owner.java
Person.java
Pet.java
PetType.java
Specialty.java
Vet.java
Pet
Owner
PetType
Vet
public Owner getOwner() {
return this.owner;
}
protected void setVisitsInternal(Set<Visit> visits) {
this.visits = visits;
}
protected Set<Visit> getVisitsInternal() {
if (this.visits == null) {
this.visits = new HashSet<Visit>();
}
return th
}
public List<V
List<Visi
ernal());
97:18
UTF-8
Java
Proposals:
birthDate : DateTime - Pet
id : Integer - BaseEntity
owner : Owner - Pet
type : PetType - Pet
visits : Set<org.springframework.samples.petclinic.model.Visi
addVisit(Visit visit) : void - Pet
clone() : Object - Object
equals(Object arg0) : boolean - Object
finalize() : void - Object
getBirthDate() : DateTime - Pet
Processes
DEV
ws-machine
SSH
Terminal
build and run
ws-machine
Terminal
build and
drwxr-xr-x 2 root root 6 Jan 19 16:
drwxr-xr-x 2 root root 6 Jan 19 16:
drwxr-xr-x 2 root root 6 Jan 19 16:
dr-xr-xr-x 317 root root 0 Mar 8 08:51 proc
drwxr-xr-x 3 user root 4096 Mar 6 18:07 projects
drwx------ 2 root root 37 Jan 19 16:33 root
drwxr-xr-x 6 root root 127 Mar 8 08:52 run
Events
Processes
Projects
Commands
Pull Request


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Mon 17:19
aaabc 〉 iPhone X
aaabc | Build aaabc: Succeeded | Today at 14:52
Simulator - iOS 12.0 〉 Frameworks 〉 UIKit 〉 UIDevice.h 〉 UIUserInterfaceIdiom
aaabc
aaabc
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
1 #if USE_UIKIT_PUBLIC_HEADERS || !__has_include(<UIKitCore/UIDevice.h>)
2 //
3 // UIDevice.h
4 // UIKit
5 //
6 // Copyright (c) 2007-2017 Apple Inc. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10 #import <UIKit/UIKitDefines.h>
11
12 NS_ASSUME_NONNULL_BEGIN
13
14 typedef NS_ENUM(NSInteger, UIDeviceOrientation) {
15 UIDeviceOrientationUnknown,
16 UIDeviceOrientationPortrait, // Device oriente
17 UIDeviceOrientationPortraitUpsideDown, // Device oriente
18 UIDeviceOrientationLandscapeLeft, // Device oriente
19 UIDeviceOrientationLandscapeRight, // Device oriente
20 UIDeviceOrientationFaceUp, // Device oriente
21 UIDeviceOrientationFaceDown // Device oriente
22 } __TVOS_PROHIBITED;
23
Quick Help
Summary
An interface designed for an in-car experience.
Declaration
UIUserInterfaceIdiomCarPlay
Open in Developer Documentation
Auto
Filter
Filter

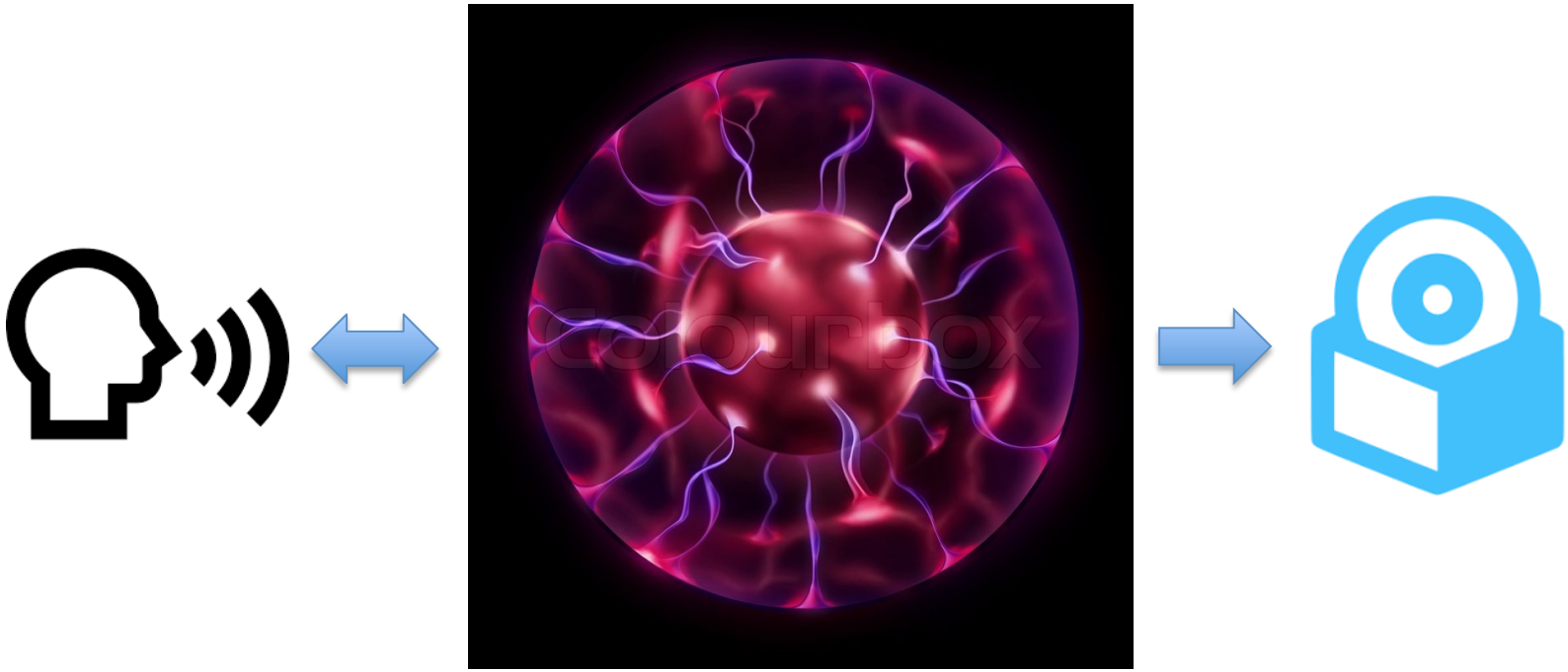

CompilationAndLinking - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM INCREDIBUILD TOOLS TEST ARCHITECTURE RESHARPER ANALYZE WINDOW HELP
Local Windows Debugger
Debug
foo.cpp
foo.h
main.cpp*
(Global Scope)
main()
#include <iostream>
#include <cassert>
using namespace std;
#include "foo.h"
#if ABC
void foo() {}
#endif
#define MUL(a,b) a*b
int main()
{
cout << "2+3=" << add(2,3)
'add' (#include "../StaticLib/foo.h")? (Alt+Enter)
cout << "Hello, C++" << endl;
getchar();
return 0;
}
100 %
Error List
Output
Find Results 1
Find Symbol Results
IncrediBuild
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'CompilationAndLinking' (2 projects)
HelloCpp
External Dependencies
Header Files
foo.h
Resource Files
Source Files
foo.cpp
main.cpp
StaticLib
External Dependencies
Header Files
foo.h
Resource Files
Source Files
foo.cpp
Solution E...
Properties
Team Expl...
Class View
Server Explorer
Toolbox
Ready
Ln 15
Col 28
Ch 28
pluralsight



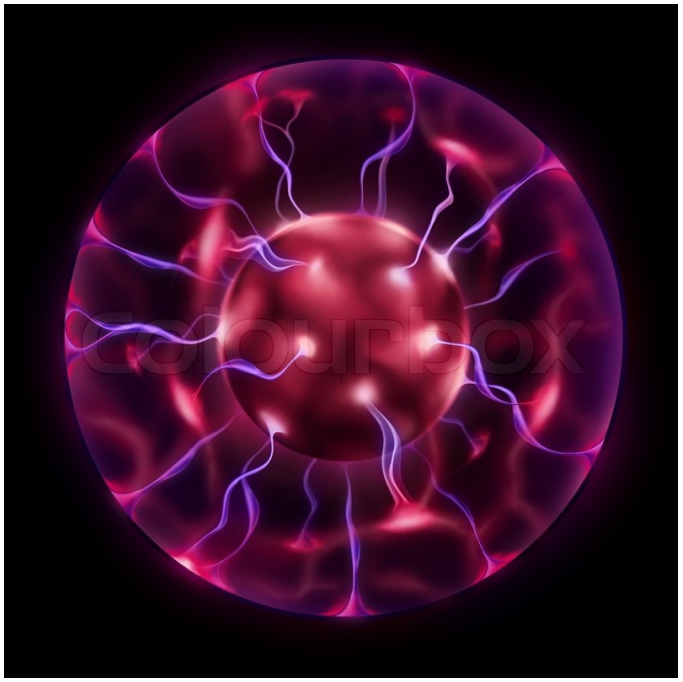
# Can we move beyond “coding”?



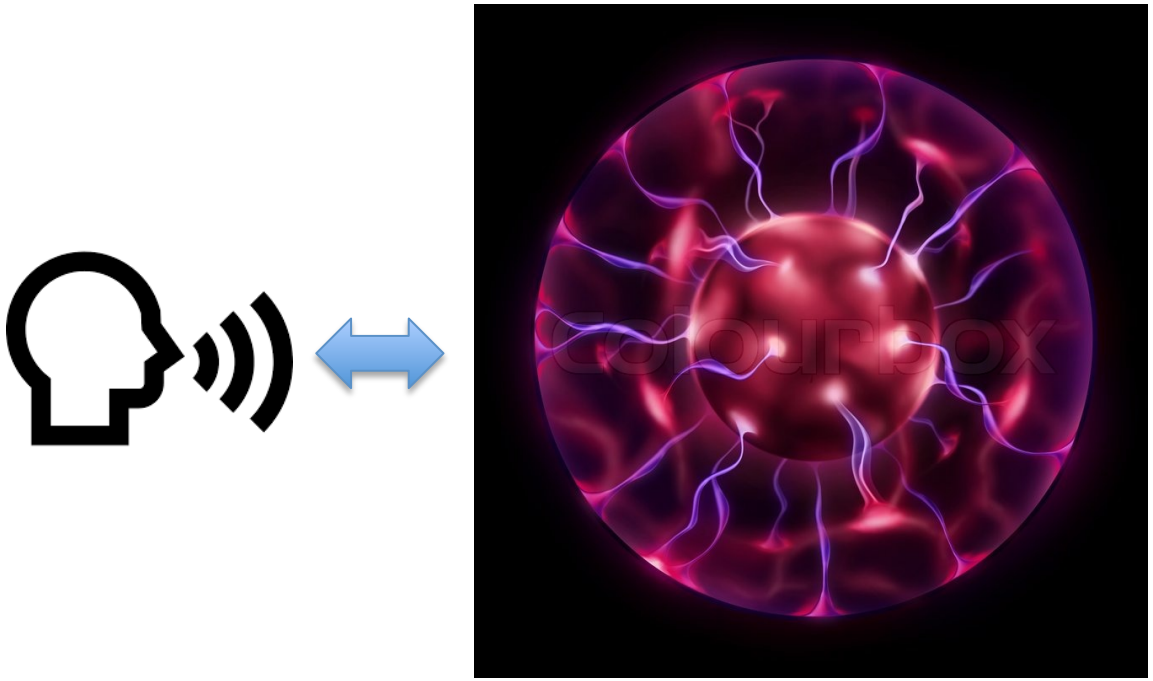
your wish?



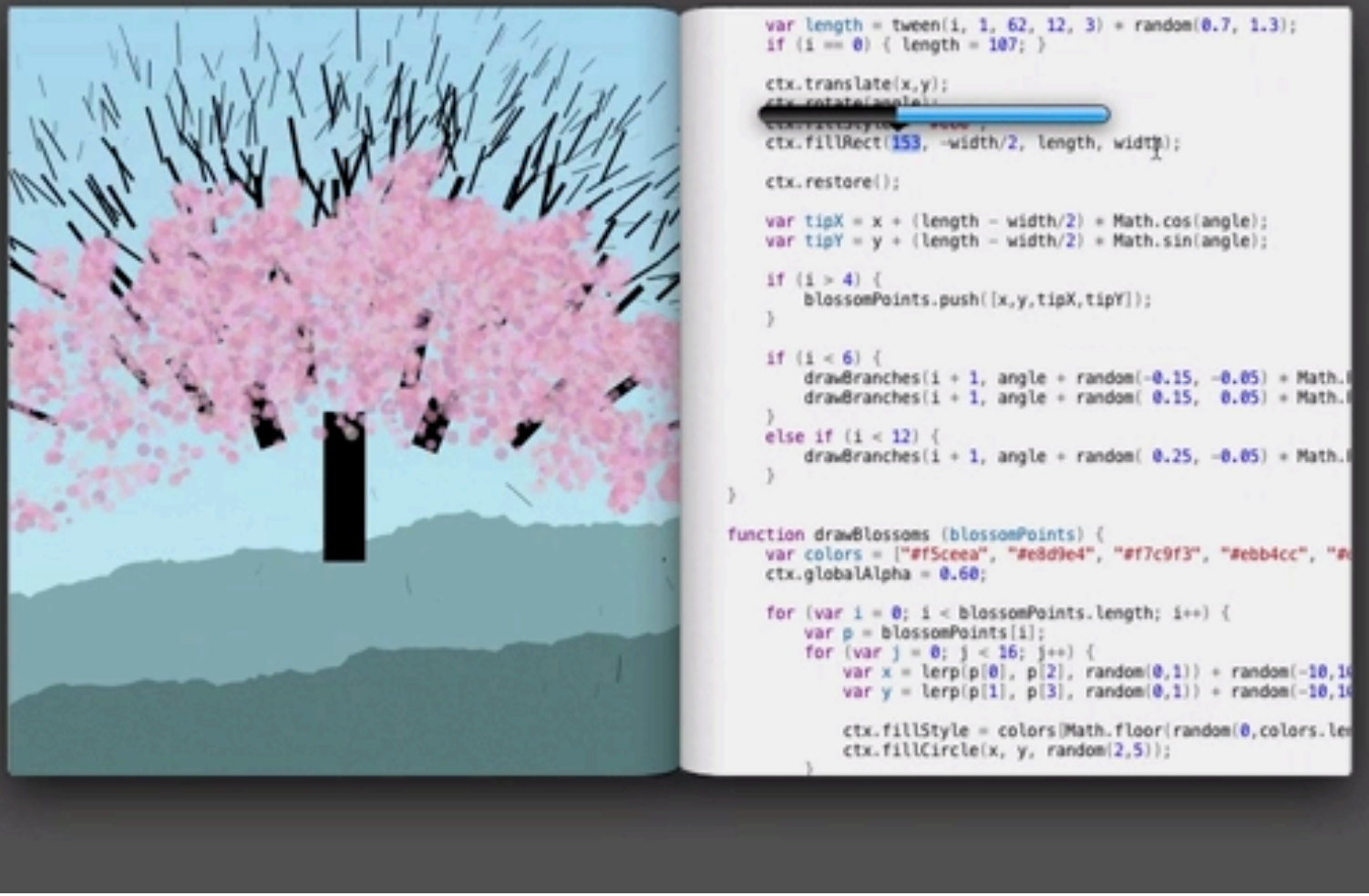
# Similarity is key



# Communicating the wish the hardest

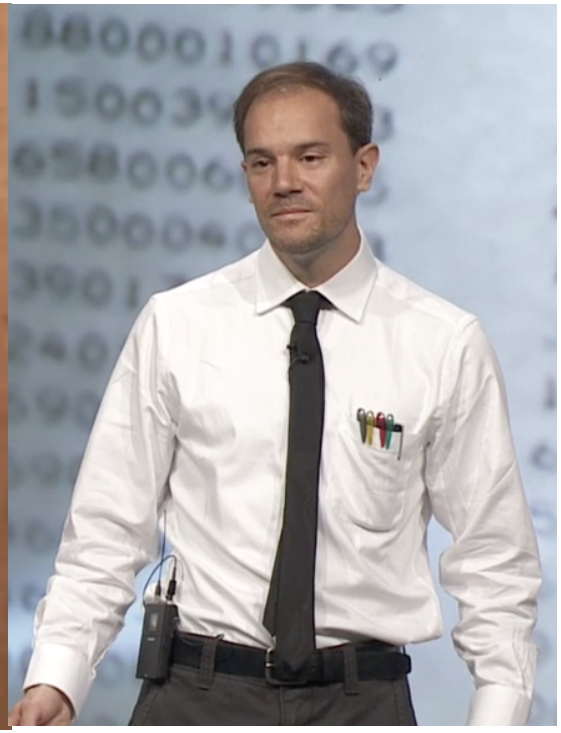


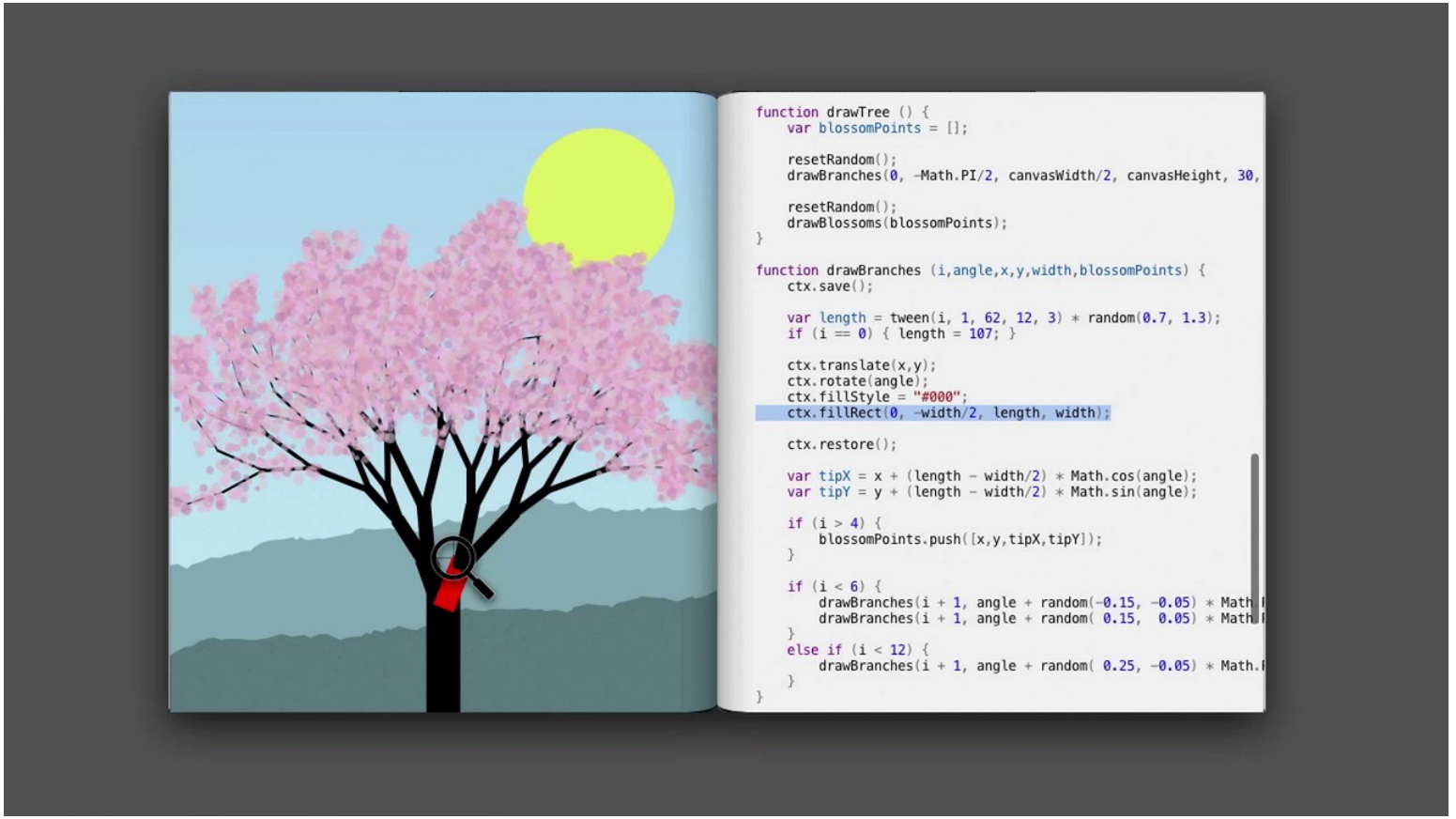
```
function drawTree () {
    var blossomPoints = [];

    resetRandom();
    drawBranches(0, -Math.PI/2, canvasWidth/2, canvasHeight, 30,

    resetRandom();
    drawBlossoms(blossomPoints);
}

function drawBranches (i,angle,x,y,width,blossomPoints) {
    ctx.save();

    var length = tween(i, 1, 62, 12, 3) * random(0.7, 1.3);
    if (i == 0) { length = 107; }

    ctx.translate(x,y);
    ctx.rotate(angle);
    ctx.fillStyle = "#000";
    ctx.fillRect(0, -width/2, length, width);

    ctx.restore();

    var tipX = x + (length - width/2) * Math.cos(angle);
    var tipY = y + (length - width/2) * Math.sin(angle);

    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    if (i < 6) {
        drawBranches(i + 1, angle + random(-0.15, -0.05) * Math.
        drawBranches(i + 1, angle + random( 0.15,  0.05) * Math.
    }
    else if (i < 12) {
        drawBranches(i + 1, angle + random( 0.25, -0.05) * Math.
    }
}
```



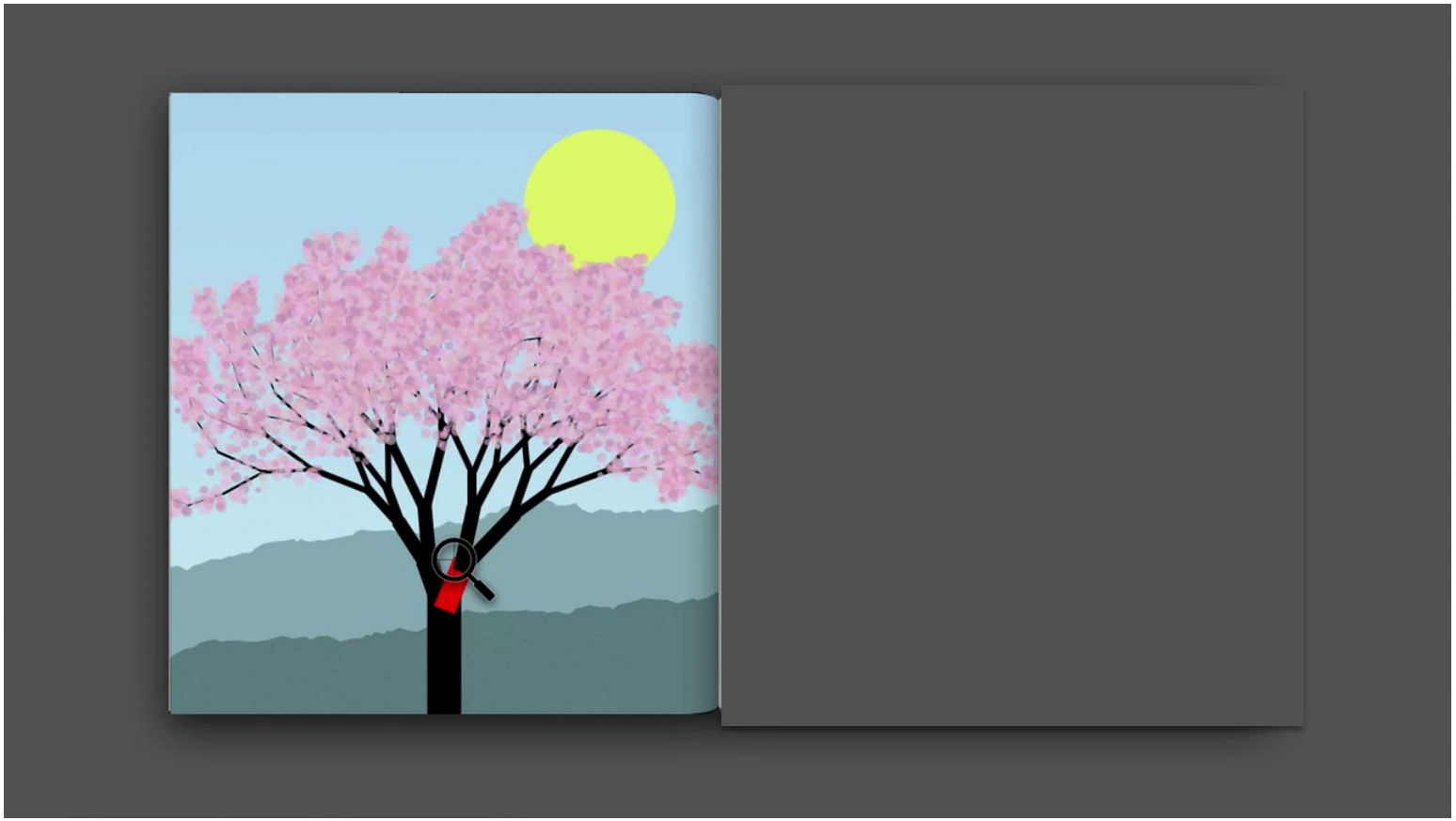
```
    var length = tween(i, 1, 62, 12, 3) * random(0.7, 1.3);
    if (i == 0) { length = 107; }

    ctx.translate(x,y);
    ctx.rotate(angle);
    ctx.fillStyle = '#000';
    ctx.fillRect(153, -width/2, length, width);

    ctx.restore();

    var tipX = x + (length - width/2) * Math.cos(angle);
    var tipY = y + (length - width/2) * Math.sin(angle);

    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    if (i < 6) {
        drawBranches(i + 1, angle + random(-0.15, -0.05) * Math.
        drawBranches(i + 1, angle + random( 0.15,  0.05) * Math.
    }
    else if (i < 12) {
        drawBranches(i + 1, angle + random( 0.25, -0.05) * Math.
    }
}

function drawBlossoms (blossomPoints) {
    var colors = ["#f5ceea", "#e8d9e4", "#f7c9f3", "#ebb4cc", "#
    ctx.globalAlpha = 0.60;

    for (var i = 0; i < blossomPoints.length; i++) {
        var p = blossomPoints[i];
        for (var j = 0; j < 16; j++) {
            var x = lerp(p[0], p[2], random(0,1)) + random(-10,1
            var y = lerp(p[1], p[3], random(0,1)) + random(-10,1

            ctx.fillStyle = colors[Math.floor(random(0,colors.le
            ctx.fillCircle(x, y, random(2,5));
        }
```

# Bret Victor: Inventing on Principle





```
function drawTree () {
    var blossomPoints = [];

    resetRandom();
    drawBranches(0, -Math.PI/2, canvasWidth/2, canvasHeight, 30,

    resetRandom();
    drawBlossoms(blossomPoints);
}

function drawBranches (i,angle,x,y,width,blossomPoints) {
    ctx.save();

    var length = tween(i, 1, 62, 12, 3) * random(0.7, 1.3);
    if (i == 0) { length = 107; }

    ctx.translate(x,y);
    ctx.rotate(angle);
    ctx.fillStyle = "#000";
    ctx.fillRect(0, -width/2, length, width);

    ctx.restore();

    var tipX = x + (length - width/2) * Math.cos(angle);
    var tipY = y + (length - width/2) * Math.sin(angle);

    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    if (i < 6) {
        drawBranches(i + 1, angle + random(-0.15, -0.05) * Math.
        drawBranches(i + 1, angle + random( 0.15,  0.05) * Math.
    }
    else if (i < 12) {
        drawBranches(i + 1, angle + random( 0.25, -0.05) * Math.
    }
}
```



Goal is the object, not the code

Can we directly
manipulate & explore the object
to express the “wish”?

Can we directly
manipulate & explore the object
to express the “wish”?

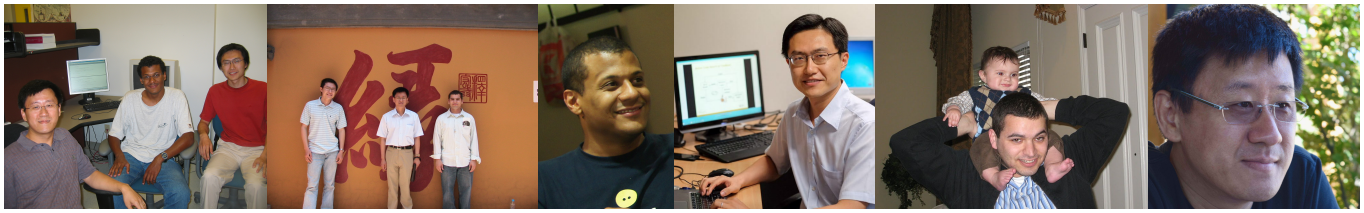
Perhaps via visualization & virtual reality?

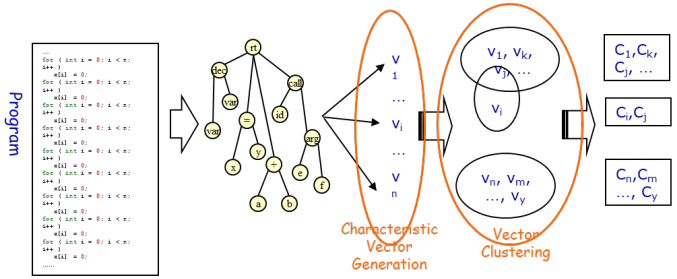
We had fun!

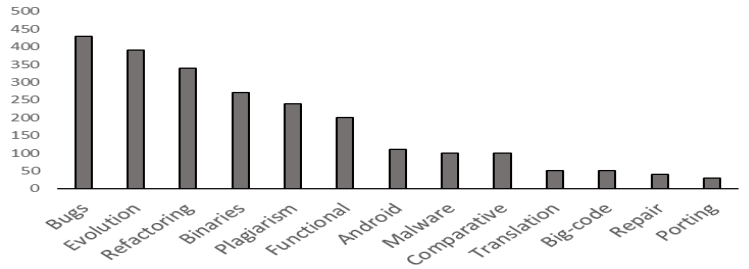
We worked hard!

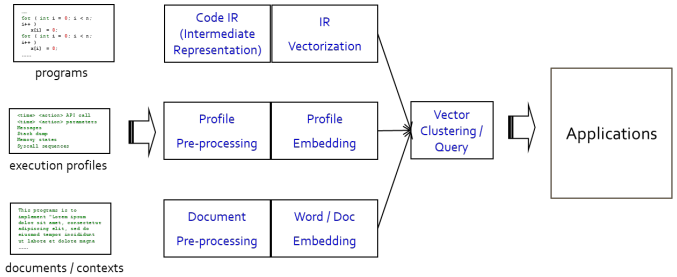
We are grateful!

Software is about similarity!




Program
Characteristic Vector Generation
Vector Clustering


500
450
400
350
300
250
200
150
100
50
0
Bugs
Evolution
Refactoring
Binaries
Plagiarism
Functional
Android
Malware
Comparative
Translation
Big-code
Repair
Porting


programs
execution profiles
documents / contexts
Code IR (Intermediate Representation)
IR Vectorization
Profile Pre-processing
Profile Embedding
Document Pre-processing
Word / Doc Embedding
Vector Clustering / Query
Applications


your wish?

# Thank you!